SI539 in Transition

SI539 - Charles Severance

It is OK and expected to feel a little lost for a short while.

Schedule Update

February 6: Lenz 1 + 2

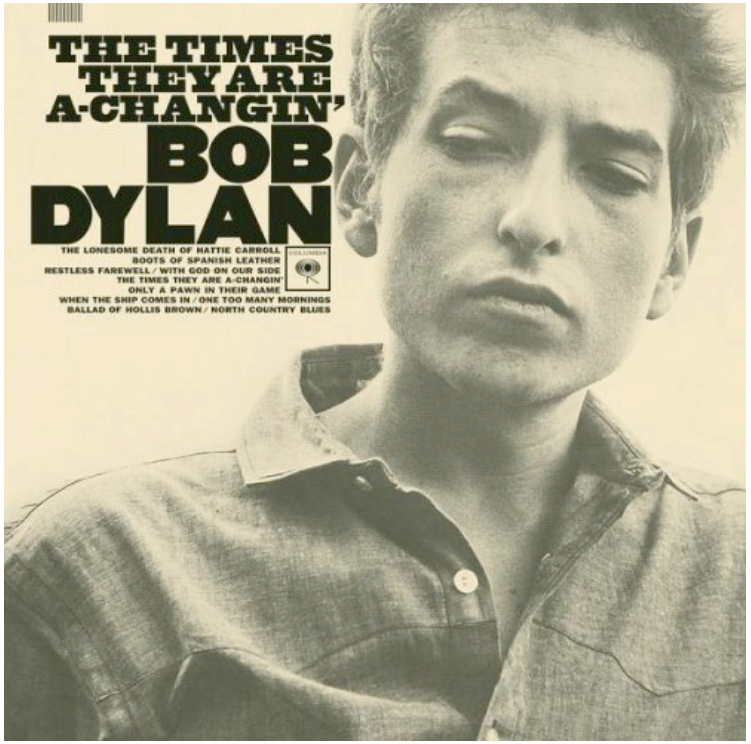
February 13: Lenz 4 + 3

February 20: Lenz 3 + 4 + Exam Review

February 27: Spring Break

March 3: Written Midterm

March 12: Practical Midterm - Lenz 5



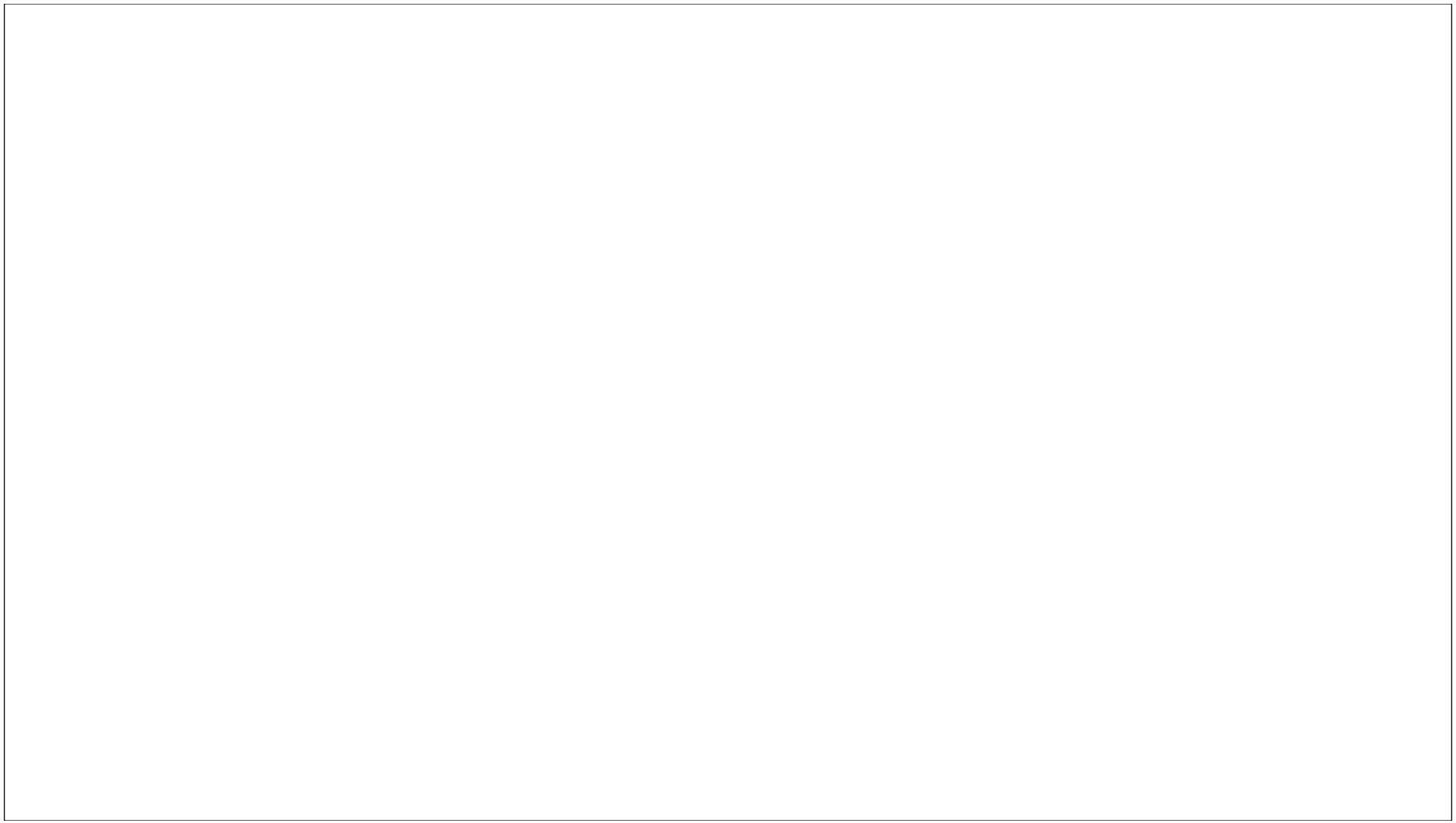
http://en.wikipedia.org/wiki/The_Times_They_Are_a-Changin'
http://www.youtube.com/watch?v=IZ_XwLSN45I

So Far....

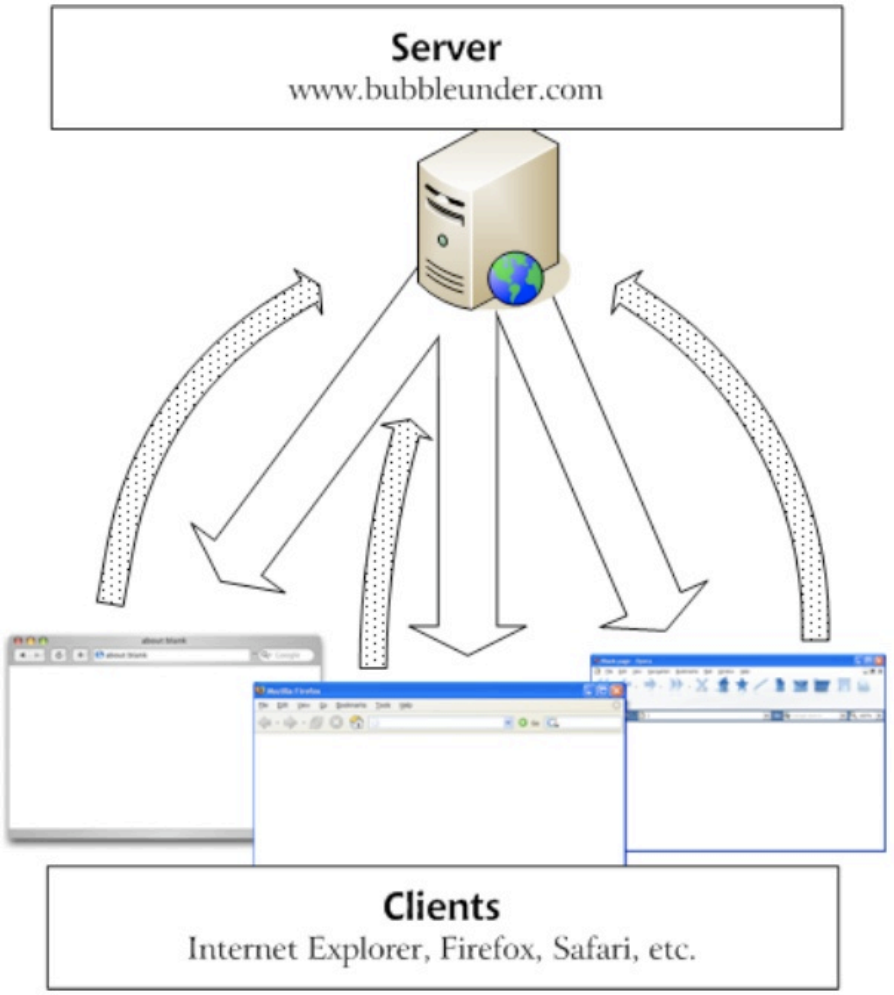
- We have studied how the Web Browser works
  - How to make web pages look nice
  - How to make web pages comply with W3C standards
  - How to make web pages accessible
  - The syntax of HTML and CSS

Up Next...

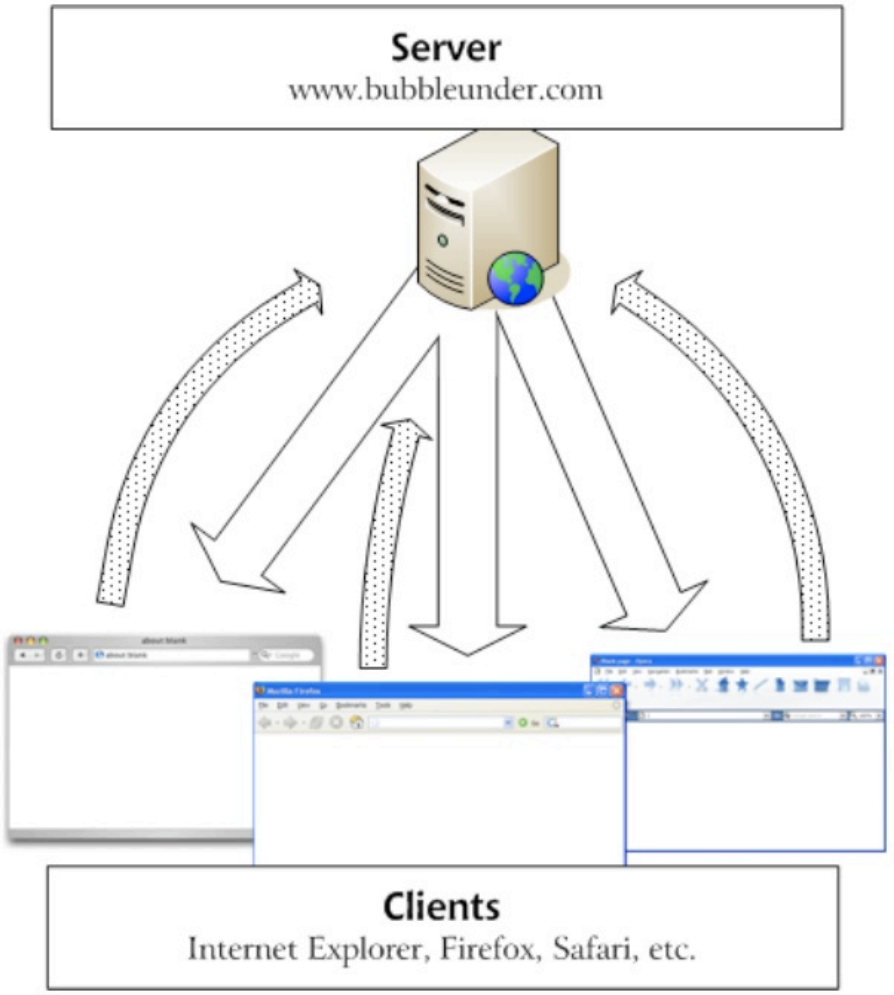
- We are going to learn how a Web Server works
  - We are going to learn how to *make* web pages using programming
  - We are going to learn how to store and retrieve data in web pages in a server.
  - We are going to learn to program - to write code
  - We are going to learn to model data and build databases



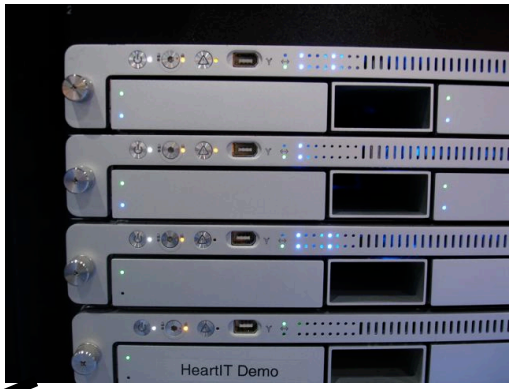
Part I


Server
www.bubbleunder.com
Clients
Internet Explorer, Firefox, Safari, etc.

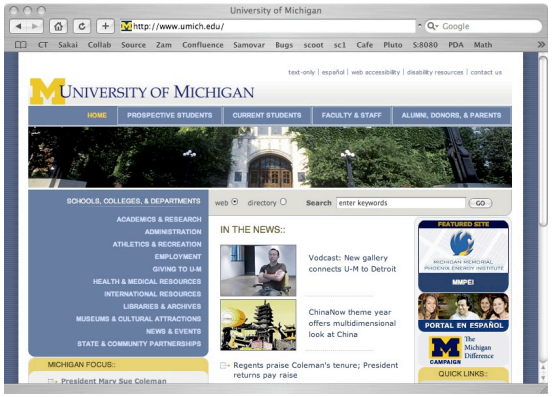
Part II


Server
www.bubbleunder.com
Clients
Internet Explorer, Firefox, Safari, etc.

The big picture...

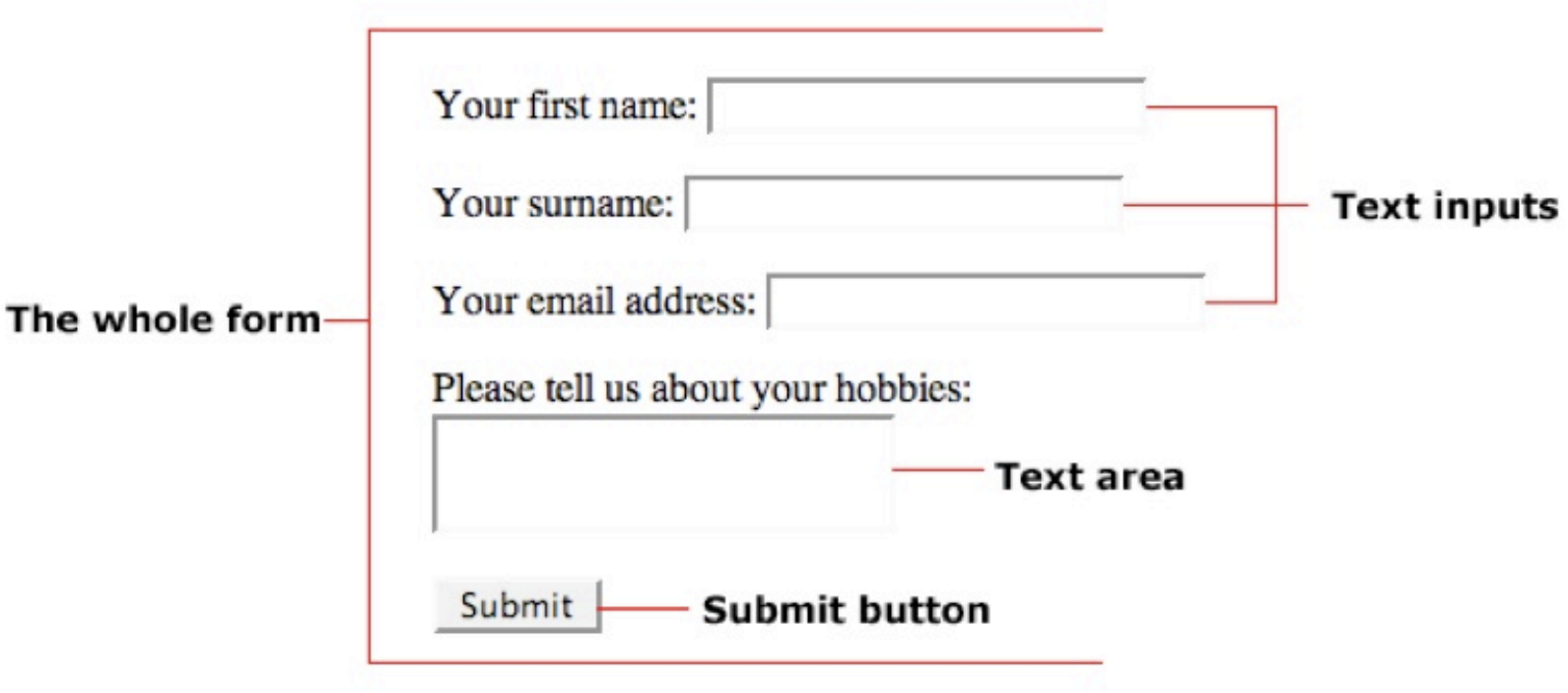



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>University of Michigan</title>
....
```



A web server produces HTML which is handed to a browser which needs to lay it out in a blink of an eye and have it pixel perfect as god as a print brochure.

Forms - Input on the Web



Forms Need Servers

- Forms effectively gather data from the user and "submit" it to a web page on a server

http://en.wikipedia.org/wiki/Common_Gateway_Interface

Using Forms Without Servers

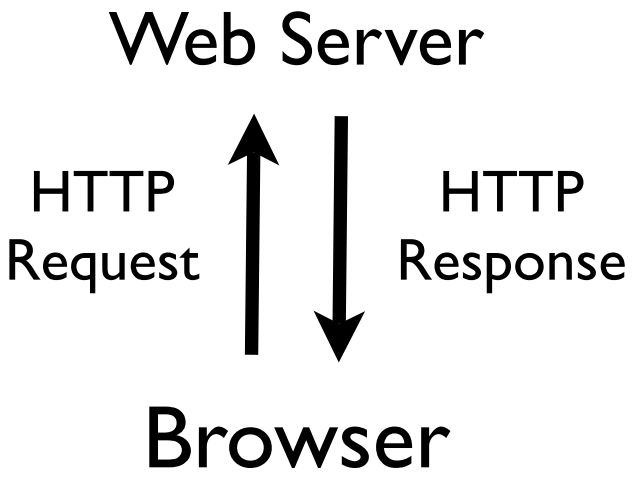
- Submitting form data works without a server - the browser moves from static page to static page
- But the data from the forms is neither saved nor is it usable

<nerdy-stuff>

Getting Data From The Server

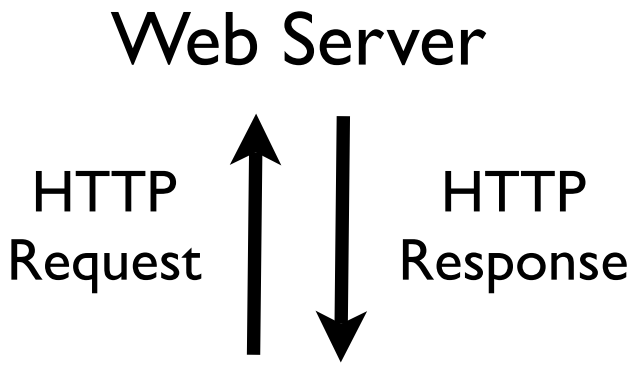
- Each the user clicks on an anchor tag with an href= value to switch to a new page, the browser makes a connection to the web server and issues a "GET" request - to GET the content of the page at the specified URL
- The server returns the HTML document to the Browser which formats and displays the document to the user.

HTTP Request / Response Cycle



http://www.oreilly.com/openbook/cgi/ch04_02.html

HTTP Request / Response Cycle

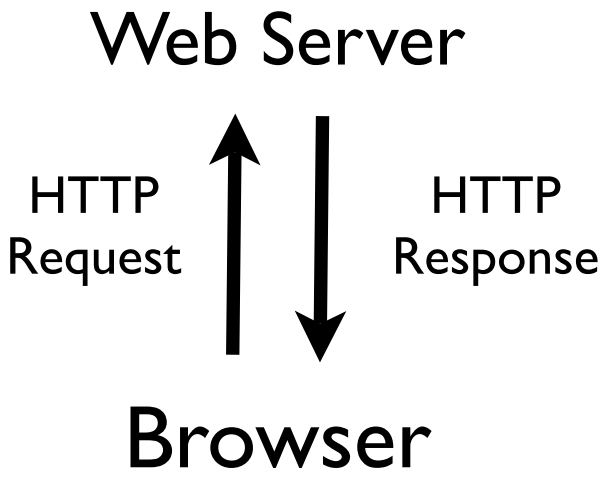


GET /index.html
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14

```
<head> .. </head>
<body>
<h1>Welcome to my
   application</h1>
 ....
</body>
```

http://www.oreilly.com/openbook/cgi/ch04_02.html

“Hacking” HTTP

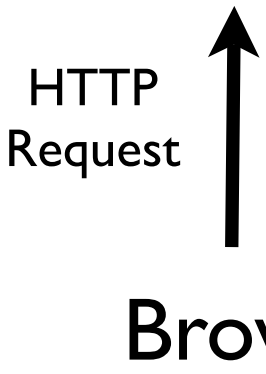



```
Last login: Wed Oct 10 04:20:19 on ttyp2
si-csev-mbp:~ csev$ telnet www.umich.edu 80
Trying 141.211.144.188...
Connected to www.umich.edu.
Escape character is '^]'.
GET /
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
 ....
```

Forms Get .vs. Post

- Two ways the browser can send parameters to the web server
  - GET - Parameters are placed on the URL which is retrieved
  - POST - The URL is retrieved and parameters are appended to the request in the the HTTP connection

Passing Parameters to The Server



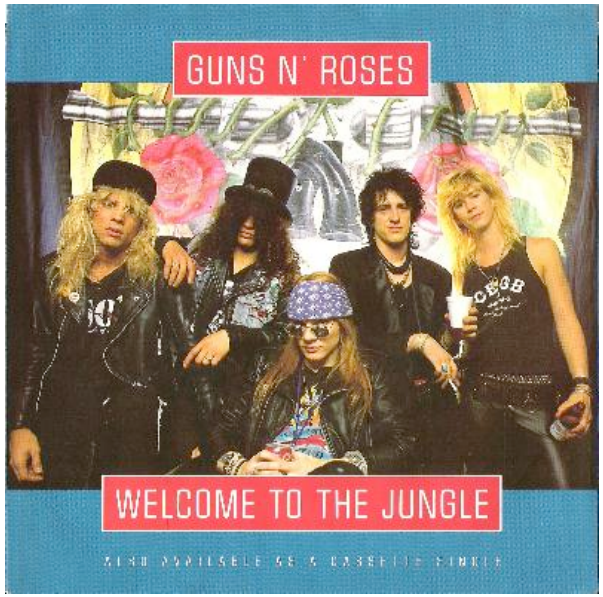

```
GET /simpleform.html?yourname=fred
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14
```

```
POST /cgi-bin/program.pl HTTP/1.0
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14
Content-type: application/x-www-form-urlencoded
Content-length: 13
yourname=fred
```

```
<input type="text" name="yourname" id="yourid" />
```

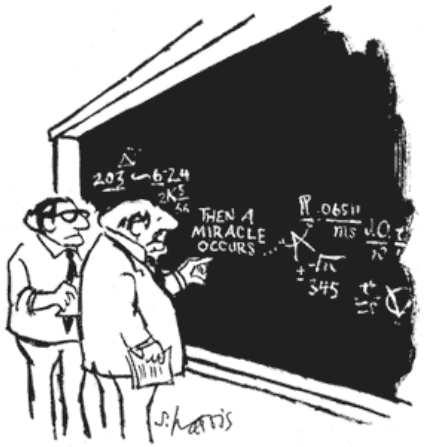
</nerdy-stuff>

Welcome to the Server



http://en.wikipedia.org/wiki/Welcome_to_the_Jungle

http://www.youtube.com/watch?v=xtXN_EHPwSg



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Web Server

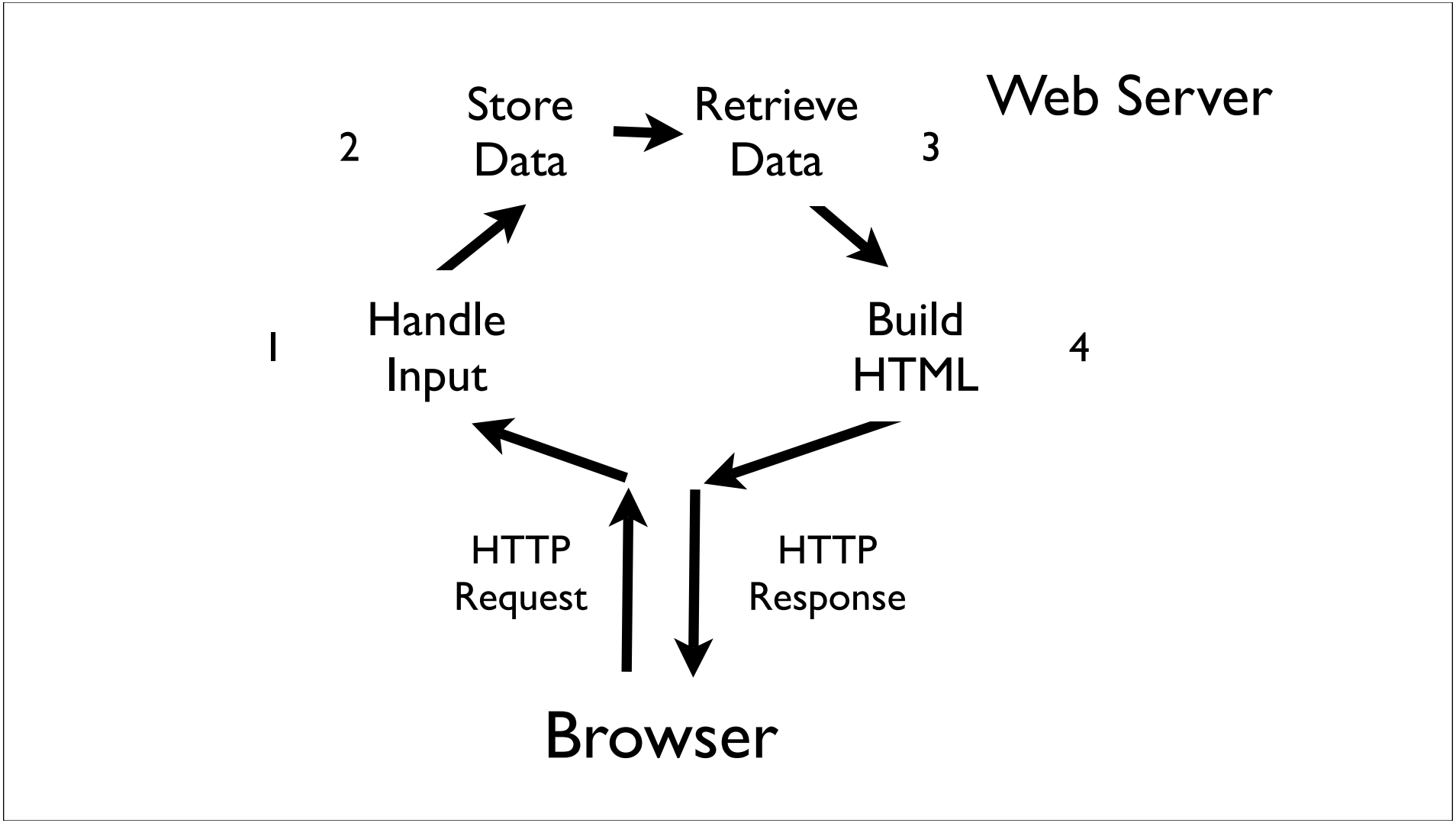
HTTP Request

HTTP Response

Browser

Tasks Inside the Server

- Process any form input - possibly storing it in a database or making some other change to the database such as a delete
- Decide which screen to send back to the user
- Retrieve any needed data
- Produce the HTML response and send it back to the browser


Web Server
2
Store Data
Retrieve Data
3
1
Handle Input
Build HTML
4
HTTP Request
HTTP Response
Browser

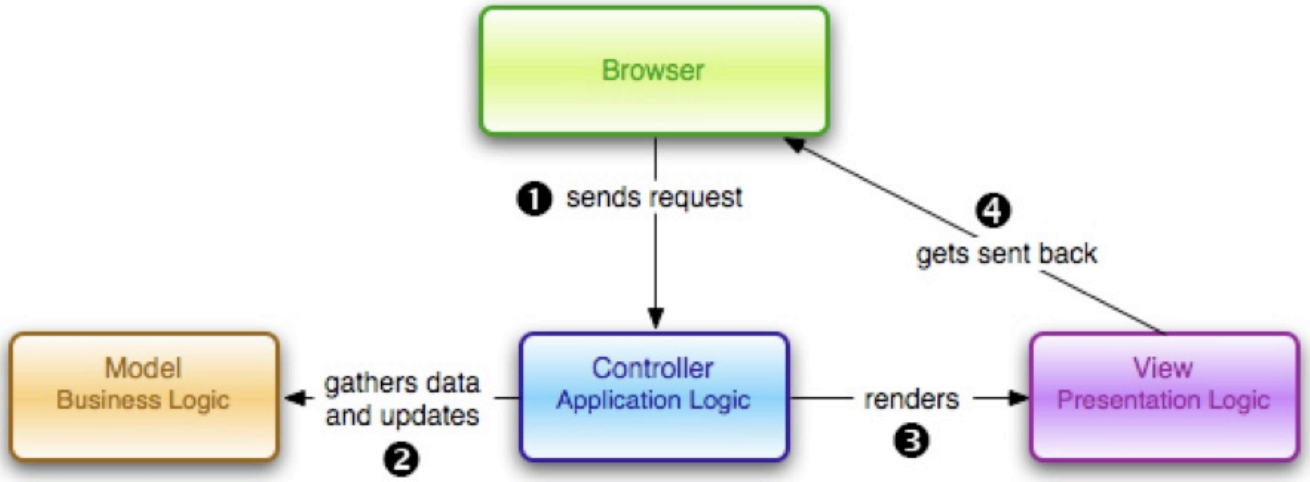
Programming Style in Servers

- There is a wide range of approaches to programming in servers
  - Write a bunch of little programs that each do all four tasks for one of many screens in the program
  - Make some rules and keep things very well separated
- Having these overriding rules (or patterns) is what we call "Architecture"

Our Architecture: MVC

- Controller - Receives each request and handles input and orchestrates the other elements
- Model - Holds the permanent data which stays long after the user has closed their web browsers
- View - Produces the HTML Response

MVC - Request - Response Cycle



We should probably call this CMCV but MVC sounds better.

Summary

- We are moving from a focus on how to interact with Web Browsers using HTML and CSS towards learning how Web Servers dynamically produce
- You will learn all new syntax - all new tools - all new patterns
- You will learn to program and data model
- Relax - you will master this as well - please use the podcasts